数理解析研究所講究録
880巻 1994年 159-166

Variable coefficient Explicit Runge-Kutta Methods

鹿児島大学(理) 中島正治 (Masaharu. Nakashima)

Abstract

This paper deals with the construction of a class of A-stable explicit Runge-Kutta methods, the present methods can not be written down for a system simply by replacing scalars by vectors, but are still compoment applicable to a system. Finally, some numerical tests justifing the results are present.

1. Introduction

The present paper is concerned with the numerical integration of stiff system of ordinary differential equation:

$$y' = f(x,y), \; y(x_0) = y_0 . \tag{1.1}$$

A basic difficulty in the numerical solution of stiff system is the satisfying of the requirement of stability, from the restriction of stability, implicit type methods have been present and some explicit methods imposed the stability conditions have derived, however there still remain stability problem for the explicit methods, so it is the purpose of the present paper to derive the explicit A-stable Runge-Kutta methods with respect to the model equation. The outline of this paper is as follows:

In § 2, We consider two-stage of order one Runge-Kutta methods.

In § 3, we proposed some numerical tests.

2. Derivation of the formulae

Consider the r-stage explicit Runge-Kutta methods:

$$y_{n+1} = y_n + h \sum_{i=1}^{r} b_i k_i \quad , \tag{2.1}$$

$$k_1 = f(x_n, y_n),$$

$$k_i = f(x + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j),$$

$$c_i = \sum_{j=1}^{i-1} a_{ij} \quad (i = 2, .., r).$$

The order conditions of the R-K methods which are discussed in [1], are listing up to two order:

order 1: $\quad \Sigma b_i = 1, \quad$ (2.2)

order 2: $\quad \Sigma b_i c_i = 1/2, \quad$ (2.3)

Let us now apply the r-stage, p-th order Runge-Kutta methods (2.1) to the test equation

$$y' = \lambda y , \tag{2.4}$$

then we have the stability polynomial

$$y_{n+1} = S(z) y_n , \tag{2.5}$$

where S(z) takes the form

$$S(z) = \sum_{i=1}^{p} \frac{z^i}{i!} + \sum_{\kappa=p+1}^{r} \gamma z^{\kappa} , \qquad (z = \lambda h),$$

We shall study the stability polynomial S(z) with (p,r) = (1,2).

2.1: first order Runge-Kutta formula with two-stage

From (2.5) the numerical processes is

$$y_{n+1} = (1 + z + b_2 a_{21} z^2) y_n , \tag{2.6}$$

here we assume that $b_2 a_{21}$ has the form:

$$b_2 \; a_{21} \;=\; \frac{\sum_{i=0}^{m} \tilde{b}_i z^i}{\sum_{i=0}^{s} \tilde{a}_i z^i} \quad .$$

From the stability condition it is required

$$m \leq s - 1 \quad .$$

We may choice the integer s in any number, in this paper we study only the case s = 2:

$$b_2 \; a_{21} \;=\; \frac{\delta + \rho z}{\alpha + \beta z + \gamma z^2} \, . \tag{2.7}$$

Putting (2.8) into (2.7) yields

$$y_{n+1} = \frac{\alpha + (\alpha + \beta)z + (\gamma + \beta + \delta)z^2 + (\gamma + \rho\;)z^3}{(\alpha + \beta h + \gamma z^2)} y_n , \tag{2.8}$$

where the undetermined parameters $\alpha,\beta,\gamma,\delta$ and ρ must be chosen so that the condition of A-stability is satisfied. we present some A-stable algorithms.

Case (1): If ,for example, we chosse those parameters to

$$\alpha = 1, \; \beta = -1 \;, \gamma = 0, \; \delta = 1 \text{ and } \; \rho = 0 \;,$$

(2.8) reduces to

$$b_2 \; a_{21} \;=\; \frac{1}{1 - z} \;, \tag{2.9}$$

and

$$y_{n+1} = \frac{1}{1 - z} \; y_n \;,$$

which is A-stable algorithm.

Recalling that $b_2 a_{21}$ is a free parameter, we may set $b_2 a_{21}$ by

$$b_2 \; a_{21} \;=\; \frac{y_n}{y_n - \; h \; z_1} \tag{2.10}$$

where $z_1 = f(x_n, y_n)$.

We find easily that, when we apply test function (2.5) to (2.10), (2.9) may replacy by (2.10).

Soving order conditions (2.2) and stability condition (2.10), we have the coefficients as follow:

(A) if $0 < D_1 = \dfrac{y_n}{y_n - hz_1} \le 1/2$ then we take

$$b_2 = \frac{-y_n}{a_{21}(1 - hz_1)}, \quad b_1 = 1 - b_2. \tag{2.11}$$

with $z_1 = f(x_n, y_n)$. (a_{21}:free parameter)

(B) if $D_1 < 0$, or $D_1 \ge 1/2$ then we take

$$b_2 = \frac{1}{2}, \quad a_{21} = \frac{-y_n}{b_2\ (1 - h\ z_1)}, \quad b_1 = 1 - b_2. \tag{2.12}$$

with $z_1 = f(x_n, y_n)$.

Case (11): If, for example, we take

$\alpha = 1,\ \beta = -1,\ \gamma = 1,\ \delta = 1$ and $\rho = -1$

then (2.8) reduces to

$$b_2\ a_{21} = \frac{1 - z}{1 - z + z^2}, \tag{2.13}$$

and the numerical processes (2.9) has the form

$$y_{n+1} = \frac{1}{1 - z + z^2}\ y_n,$$

which ia A-stable algorithm. As the same reason of case (1), we may replace (2.13) by

$$b_2\ a_{21} = \frac{y_n - z_1}{y_n - h\ z_1 + h\ z_2} \tag{2.14}$$

with $z_1 = f(x_n, y_n)$, $z_2 = f(x_n + h, y_n + h\ z_1/2)$.

Solving order condition with (2.14), we have the coefficients as follow:

(A) if $0 \le D_2 = \dfrac{y_n - z_1}{y_n - hz_1 + hz_2} \le \dfrac{1}{2}$ then we take

$$b_2 = \frac{-y_n}{a_{21}(y_n - hz_1 + hz_2)}, \quad b_1 = 1 - b_2. \tag{2.15}$$

with $z_1 = f(x_n, y_n)$, $z_2 = f(x_n + h/2, y_n + hz_1/2)$,

(a_{21}:free parameter)

(B) if $D_2 < 0$ or $D_2 > \dfrac{1}{2}$ then we take

$$b_2 = \frac{1}{2}, \quad a_{21} = \frac{-y_n}{b_2\ (y_n - hz_1 + hz_2)}, \quad b_1 = 1 - b_2 \tag{2.16}$$

with $z_1 = f(x_n, y_n)$, $z_2 = f(x_n + h/2, y_n + hz_1/2)$.

3. Numerical Examples

In order to test the method (2.1), we wish to present some numerical results. The described methods are programmed in FORTRAN and run on the Personal Computer 9801RA(NEC). The computations are done in double precision.

(1) $y' = -1000y$, $y(0) = 1$,

(2) $Y' = A\,Y$, $Y(0) = (1,1,1)$,

with

$$A = \begin{pmatrix} -0.1 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -120 \end{pmatrix},$$

(3) $Y' = A\,Y$, $Y(0) = (1,1,1)$,

with

$$A = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & -50 & 0 \\ 0 & 0 & -120 \end{pmatrix},$$

Table 1

Result using (2.11) & (2.12) with $h = 1/2^3$ and $h = 1/2^6$

Absolute error

Problem 1

x	0.125	0.500..	1
$h=1/2^3$	0.793E-2	0.396E-8	0.157E-16
$h=1/2^6$	0.171E-9	0	0

Problem 2

($h = 1/2^6$)

x	y_1	y_2	y_3
0.0625	0.484E-5	0.553E-1	0.140E-1
0.5	0.371E-4	0.946E-8	0.210E-14
1.0	0.706E-4	0.898E-16	0.443E-29

Problem 3

(h= $1/2^6$)

x	y_1	y_2	y_3
0.0625	0.491D-5	0.055E+0	0.014D+0
0.5	0.411D-4	0.946D-8	0.210D-14
1.0	0.864D-4	0.898D-16	0.443D-29

Finally we consider a variable step algorithm. Let $y_n^{(i)}$ and $\tilde{y}_n^{(i)}$ denotes the approximation to the i-th component at x = x_n using step size h and $h/2$ respectively.

Defining

$$EST = \| y_n^{i} - \tilde{y}_n \| = \max_{1\le i\le S} | y_n^{(i)} - \tilde{y}_n^{(i)} | .$$

We use the following step size control policy for a given local accuracy requirement ε.

1. If EST > ε, reject the solution and half the step size h,
2. If $\varepsilon/50$ < EST <ε, accept the solution and keep the step size h fixed.
3. If EST < $\varepsilon/50$, accept the solution and double the step size h.

To test our automatic step control policy, we consider the problem (II) and (III) with ε = 0.1E-4. and the initial step size h=1/16 .

Absolute error

Problem 2

x	number of steps	y_1	y_2	y_3
0.00244..	12	0.588D-8	0.685D-3	0.050D-1
0.051	112	0.122E-6	0.124E-2	0.282E-3
0.107	255	0.255E-6	0.588E-3	0.687E-6

Problem 3

x	number of steps	y_1	y_2	y_3
0.0024...	12	0.603E-8	0.685E-3	0.050E-1
0.051	112	0.127E-6	0.124E-2	0.282E-3
0.107	212	0.249E-6	0.209E-3	0.147E-5

References

[1] Butcher, J. C.(1985): The Numerical Analisis of Ordinary Differential Equations, John Wiley and Sons.

[2] van der Houwen, P. J.(1973): Construction of Integration Formulas for initial Value Problems, North-Holland, Amsterdam.

[3] Nakashima, M: On the Stability Of Variable-Formula Explicit Pseudo Runge-Kutta Method. Intern. Confere. 1992 at Benin Univ.

[4] ————: Efficient Explicit Runge-Kutta Methods for Stiff Systems , To appear in Rep. Fac. Sci., Kagoshima Univ.